CS 61C Fall 2022 — Cache Performance

Discussion 10

1 Pre-Check

This section is designed as a conceptual check for you to determine if you conceptually understand and have any misconceptions about this topic. Please answer true/false to the following questions, and include an explanation:

1.1 In a write-back cache, writes are generally slower than a write-through cache.

1.2 In a fully associative cache, the number of index bits is always zero.

1.3 For the same cache size and block size, a 4-way set associative cache will have fewer index bits than a direct-mapped cache.

1.4 Using N-way set associative caches can help balance between hardware complexity and cache performance.

1.5 Increasing cache size (while keeping other parameters constant) for a fully associative cache with LRU replacement will never increase miss rate (for the same memory access pattern).

1.6 Adding another level of caches will never increase AMAT.

1.7 The miss penalty for the last level of caches is equal to the access time of main memory.

2 Cache Associativity

In the previous discussion, we primarily focuses on Direct-Mapped caches, in which blocks map to specifically one slot in our cache. This is good for quick replacement and finding out block, but not good for spatial efficiency!

This is where we bring associativity into the matter. We define associativity as the number of slots a block can potentially map to in our cache. Thus, a Fully-Associative cache has the most associativity, meaning every block can go anywhere in the cache. Our Direct-Mapped cache, on the other hand, has the least, being only 1-way set associative.

For an N-way associative cache, the following are true:

$$\text{N} * \text{\# sets} = \text{\# blocks}, \quad \text{Index bits} = \log_2(\text{\# sets})$$

2.1 Here's some practice involving a 2-way set associative cache. This time we have an 8-bit address space, 8 B blocks, and a cache size of 32 B. Classify each of the following accesses as a cache hit (H), cache miss (M) or cache miss with replacement (R). For any misses, list out which type of miss it is (Compulsory, Conflict, or Capacity). Assume that we have an LRU replacement policy (in general, this is not the case).

Address	T/I/O	Hit, Miss, Replace
`0b0000 0100`		
`0b0000 0101`		
`0b0110 1000`		
`0b1100 1000`		
`0b0110 1000`		
`0b1101 1101`		
`0b0100 0101`		
`0b0000 0100`		
`0b1100 1000`		

2.2 What is the hit rate of our above accesses?

3 Writes

When it comes to writing data to cache memory, there are multiple write policies to consider that offer different options when building our system. Some of them you might encounter are:

1. **Write-through**: In this policy, when we have a write we write to both the cache and the memory. This is the case for every write, so the main memory always has the updated data. This is simple to implement, but writing to main memory every single time is slow.
2. **Write-back**: On a write, the data is only updated/written in the cache. The main memory only receives the data upon eviction. This means the cache has more up to date data most of the time. While this is faster as there is less

accesses to main memory, it is harder to implement as we have to include more overhead, such as dirty bits and so on.

3. **Write-around**: Data is only written to main memory, and whenever we do so we automatically invalidate the old data in the cache.

Another thing to consider is what we do when we want to write to memory that is not in the cache, or a write miss. For that, we have 2 possible policies:

1. **Write-allocate**: On a write miss, we pull the block you missed on into the cache
2. **No write-allocate**: On a write miss, you do not pull the block you missed on into the cache. Only memory is updated. On a read cache miss, we still pull the data into the cache.

3.1 Considering the above information, lets consider a direct mapped, no write-allocate write-through cache with a capacity of 8B and a block size of 4B. Lets also assume that the memory addresses are 8 bits each. Assuming the cache is completely empty in the beginning, we make memory accesses to the following locations:

- 0x6A, Write
- 0x85, Read
- 0x6B, Read
- 0x87, Read
- 0x68, Write

With the above memory access pattern and the given cache configuration, how many times do we access the main memory?

3.2 Lets say for the same cache size, memory accesses but the only difference is that we have a no write-allocate write-back cache instead. How many memory accesses to the main memory do we have in this case?

3.3 For one last optimization, we decide to use a write-allocate cache instead. So, now we have a write-allocate, write-back cache. How many times do we access the memory now?

3.4 We discovered in our previous subparts that one set of write policies leads to less main memory accesses than the others. Out of the policies covered in class, is this the most optimal set of write policies for a cache? If so, explain why. If they are not, what trade-offs come as a result of these policies?

4 AMAT

Recall that AMAT stands for Average Memory Access Time. The main formula for it is:

$$\text{AMAT} = \text{Hit Time} + \text{Miss Rate} * \text{Miss Penalty}$$

In a multi-level cache structure, we can separate miss rates into two types that we consider for each level.

- **Global:** Calculated as the number of accesses that missed at that level divided by the total number of accesses *to the cache system.*
- **Local:** Calculated as the number of accesses that missed at that level divided by the total number of accesses *to that cache level.*

4.1 In a 2-level cache system, after 100 total accesses to the cache system, we find that the L2$ (L2 cache) ended up missing 20 times. What is the global miss rate of L2$?

4.2 Given the system from the previous subpart, if L1$ had a local miss rate of 50%, what is the local miss rate of L2$?

Suppose your system consists of:

1. An L1$ that has a hit time of 2 cycles and has a local miss rate of 20%
2. An L2$ that has a hit time of 15 cycles and has a global miss rate of 5%
3. Main memory where accesses take 100 cycles

4.3 What is the local miss rate of L2$?

4.4 What is the AMAT of the system?

4.5 Suppose we want to reduce the AMAT of the system to 8 cycles or lower by adding in a L3$. If the L3$ has a local miss rate of 30%, what is the largest hit time that the L3$ can have?